

## A Graph Partitioning Model of Congressional Redistricting

Shawn Doyle  
*Youngstown State University*

Follow this and additional works at: <https://scholar.rose-hulman.edu/rhumj>

---

### Recommended Citation

Doyle, Shawn (2015) "A Graph Partitioning Model of Congressional Redistricting," *Rose-Hulman Undergraduate Mathematics Journal*: Vol. 16 : Iss. 2 , Article 3.  
Available at: <https://scholar.rose-hulman.edu/rhumj/vol16/iss2/3>

# A GRAPH PARTITIONING MODEL OF CONGRESSIONAL REDISTRICTING

Shawn Doyle<sup>a</sup>

VOLUME 16, No. 2, FALL 2015

Sponsored by

Rose-Hulman Institute of Technology  
Department of Mathematics  
Terre Haute, IN 47803  
Email: [mathjournal@rose-hulman.edu](mailto:mathjournal@rose-hulman.edu)  
<http://www.rose-hulman.edu/mathjournal>

---

<sup>a</sup>Youngstown State University

# A GRAPH PARTITIONING MODEL OF CONGRESSIONAL REDISTRICTING

Shawn Doyle

**Abstract.** Redrawing congressional districts in the United States is a constitutionally required, yet politically controversial, task undertaken after each decennial census. Federal law requires contiguous, ‘relatively compact’ congressional districts that maintain ‘approximately equal’ population. Controversy is introduced when individual states redraw their districts, or redistrict, using partisan committees. States such as Ohio continue to redistrict with a committee appointed according to the current proportion of legislators’ political parties to the whole. When political parties have majority power in redistricting committees, they can draw districts in a way that gives their party the best chance to keep its majority representation, a process called gerrymandering. Mathematical redistricting models seek an unbiased computational approach to the problem. Rather than trust partisan committees, mathematical modeling approaches rely upon well-defined methods in computational geometry, graph theory, game theory, and other fields. Here, we discuss two such approaches. The first, given as a background for comparison, constructs Voronoi diagrams to redistrict states into convex polygons, which are generally considered ‘compact’. We give greater emphasis to a new model that discretizes a state’s population and partitions it into regions of approximately equal population. This model, our main focus, relies upon graph partitioning to achieve the desired result and uses census population data as the sole parameter in redistricting.

---

**Acknowledgements:** The author is gratefully indebted to the insightful advisement of Dr. Tom Wakefield and Dr. George Yates in this research. I would also like to thank YSU CURMath for funding this project and my research partner Eric Shehadi, whose exceptional mastery of GIS played no small part in this work.

# 1 Introduction

The United States allots 435 congressional districts among the 50 states according to population taken in the census. Each state contains at least one district, with each district having one representative, and those states with the largest populations are given the most representatives. Districts are required to be contiguous landmasses contained within their respective states. Also, districts are required by federal law to be ‘relatively compact’ and ‘approximately equal’ in population. By a provision of the Voting Rights Act of 1965, minority ‘communities of interest’ must not be divided. None of these conditions are clearly defined by the Supreme Court or existing laws.

Generating districts that satisfy federal law and eliminate partisan bias is a difficult task. By partisan bias, we mean the common abuse of redistricting by committee called gerrymandering. Most states are redistricted by a partisan committee, and this process is biased. An excellent example of modern gerrymandering is the 2010 redistricting map of the state of Ohio, shown in Figure 1. The six most populated cities in the state are denoted with circles. Note the irregular shapes of the districts, and how they divide the state’s population hubs.

Computational methods provide solutions that are unsusceptible to gerrymandering, and they have parameters that ensure approximately equal population as well as compactness of districts. Mathematicians have considered a variety of redistricting methods, including those suggested by game theory, statistical physics, graph theory, and computational geometry. In Section 2, we begin with an overview of existing methods. In Section 3, we sample a geometric approach to the problem, while Section 4 provides a new method based upon graph partitioning. Section 5 gives an application of this method to redistrict the state of Ohio with recommendations for future work. Finally, the appendix contains population counts under several redistricting plans for the state of Ohio along with their corresponding maps.

# 2 Computational Approaches to Redistricting

Landau and Su [7] provide a game-theoretical approach where Republicans and Democrats choose their constituents in a game that gives each party equal representation. This method reduces democracy to a mutual agreement among leading political parties. Chou and Li [3] use a  $q$ -state Potts model to redistrict Taipei city. In another vein entirely, Chu et al [4] use a heuristic called the Colonial Algorithm, which echoes the behavior of competing bacteria colonies in a culture dish. Their heuristic does not necessarily converge to a solution, and the time to convergence is not known. A novel approach due to Makse [9] defines communities of interest by looking at initiative voting and builds districts from these communities, which are not to be divided by federal law, as stated above.

In a similar problem, a recent article published by Boargwardt et al [1] clusters Bavarian farmland into contiguous landmasses of approximately equal value to land-leasing farmers. The geometric clustering model combines several approaches to achieve a computationally

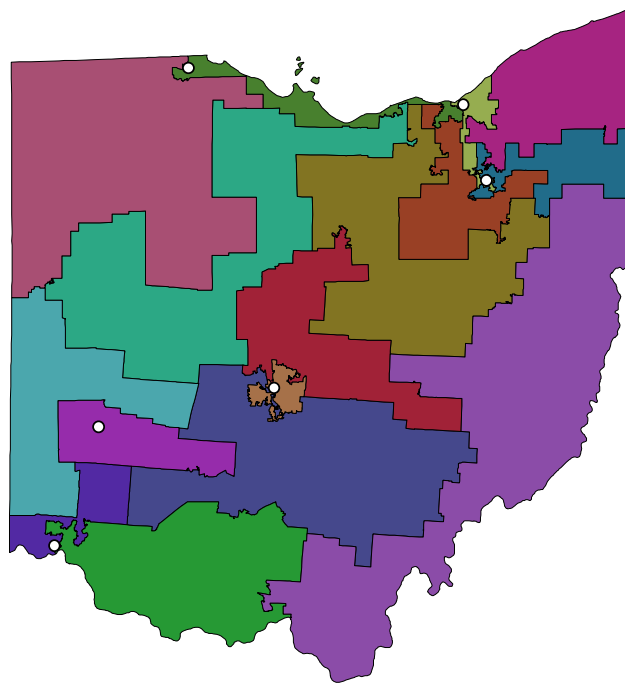


Figure 1: The 16 Ohio Congressional Districts adopted after the 2010 census

tractable solution. An excellent survey from Tasnádi [14] showcases other methods, including those that focus on public choice or graph-theoretical models. We now shift our attention to the computational geometric approach.

### 3 Geometric Approach

A Voronoi diagram is a partitioning of the Euclidean plane into convex polygons, where each polygon's points are closer to their 'generating point' than to any other generating point. Included in Figure 2 is an example of a Voronoi Diagram.

It is natural to consider convex polygons in redistricting since they are generally considered compact [2, 6, 15]; moreover, Voronoi diagrams formalize our intuition about proximity of congressional districts. To satisfy the equal population constraint, we must weight the Voronoi diagrams by population and use an algorithm that generates several polygons in a densely populated area.

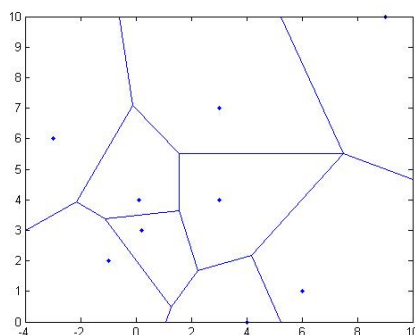


Figure 2: Voronoi diagram generated in MATLAB [10]; dots are generating points

One way to accomplish this is by strategically placing numerous generating points in cities. A team working on the 2007 COMAP mathematical modeling competition [13] used weighted Voronoi diagrams to redistrict New York state in this manner. New York City received the most generating points, Buffalo received fewer, and so on. Although this method yields compact districts, it is far from optimal. If a redistricting committee chooses where to place the generating points, it can certainly choose them to give one party an advantage.

A more formal approach is Lloyd's Algorithm [8], which computes the centroid, or center of population, of each polygon and iteratively moves each polygon's generating point to the polygon's centroid. A few iterations of this algorithm are shown in Figure 3, where the plus signs denote centroids and the red dots represent initial generating points. The algorithm always converges to a stable configuration [5]. We may expect the algorithm to give us approximately equal populations, but in reality, the populations still markedly differ. We are also left with the same problem that the COMAP team faced, namely the dependence on initial placement of points. Using Lloyd's Algorithm, it is easy to choose an area of approximately uniform population density, and yet whose population is well above or below our target value.

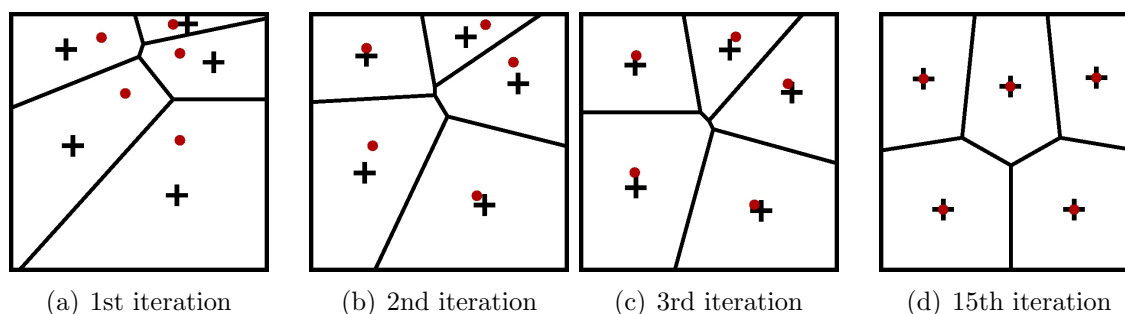


Figure 3: Lloyd's Algorithm with red generating points

Generating points must be placed in the first step of the algorithm, and it is this dependence on initial point placement that makes the method unsuitable for use in redistricting.

If one places a generating point in a rural area with approximately uniform population, then the region will not change significantly and the district will remain considerably low in population. Strategic point placement can get us closer to the desired district population, however the choice of these locations is arguably biased. It is clear that a purely geometric approach to the problem does not properly account for the uneven population distributions present in most states, so we appeal to a more population-centric method.

## 4 Graph Partitioning Method

Graph theory delivers a way to solve the redistricting problem without reliance on population demographics and Euclidean distance. Disregarding human geography and population distribution, we discretize a state into units called census block groups, the second-smallest unit that the U.S. Census Bureau analyzes. We will use these census block groups and their neighboring units to form a graph.

In particular, we consider redistricting in light of the graph partitioning problem, stated as follows: Let  $V$  be a set of vertices and  $E$  a set of edges connecting pairs of vertices. Given an undirected, vertex-weighted graph  $G = (V, E)$  with weights  $\omega(v)$  for all  $v \in V$ , we seek to partition  $G$  into  $k$  districts of approximately equal weight. The primary objective is to minimize the *edge cut*, which is the set of weighted edges whose endpoints lie in different districts. We call the sum of edge weights in an edge cut the *edge cut size*.

In terms of redistricting, the vertices represent census block groups whose weights are populations. An edge connects neighboring census block groups, and we take all initial edge weights to be 1 (our method will change this value). Our goal is to partition census block groups (the vertices), thus creating districts. Unfortunately, graph partitioning is NP-Complete, and thus an heuristic is required to provide a solution.

One heuristic, the multilevel algorithm, performs well in empirical tests, outperforming other methods both in speed and accuracy. The multilevel algorithm uses three phases — coarsening, partitioning, and uncoarsening. The coarsening phase involves recursively contracting, or collapsing, edges of the graph while preserving its basic structure. Hence we can reduce the number of vertices by a power of two, and allow more time for computationally intensive tasks. Once a chosen stopping criterion is met, the graph is partitioned into the required number of districts. Uncoarsening the graph will project this cut onto the original graph and attempt to find a better partition by swapping vertices among different districts. This is done while maintaining a pre-specified population imbalance, and again, minimizing the edge cut.

### 4.1 Coarsening

If we would like to partition graphs with millions of vertices, then it would be ideal to contract edges, making it simpler to partition. The question becomes, which edges do we contract? Finding a *maximal matching* of the graph identifies these edges with minimum computing power.

A *matching* is a set of edges where no edge pair shares a common vertex. If by adding an edge we no longer have a matching, we call the matching *maximal*. There are several algorithms for finding a maximal matching [11]; here, we use a greedy algorithm. The greedy algorithm builds an ordered list of edges using the *coarsening coefficient*

$$C(uv) = \frac{\omega(u)\omega(v)}{\omega(u \leftrightarrow v)}$$

for every edge connecting vertices  $u$  and  $v$ , denoted here by  $u \leftrightarrow v$ . A random matching would be faster to compute, but unlike the coarsening coefficient, it does not guarantee any level of quality. We now work through a simple graph partitioning problem involving a graph with seven vertices and unit edge weights.

**Example.** Consider the graph in Figure 4, where vertices are conveniently labeled with their corresponding weights (for example, vertex 2 has weight 2). Suppose that we would like to partition it into two districts, denoted  $D_1$  and  $D_2$ . First, we list the edges with their coarsening coefficients in order of increasing magnitude, as in Table 1. After constructing the table, we begin at the top of the list and work our way down, adding edges to the matching. Recall that an edge is added if it does not share a vertex with any other edge in the matching. Hence, after the inclusion of  $1 \leftrightarrow 2$ , the next edge included is  $3 \leftrightarrow 4$ .

Once we have considered every edge in the table, our matching is maximal. The maximal matching, denoted in red, is listed in Table 2 and shown in Figure 5. These edges are now contracted, so incident vertices in the matching are combined into a single vertex, and edges connecting vertices are contracted into single edges with greater weight. For example, in Figure 6, vertices 1 and 2 are contracted into a vertex labeled  $1+2$ , and the weight of this new vertex is 3.

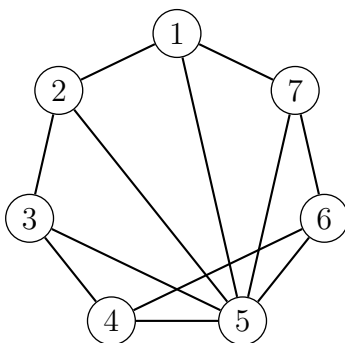


Figure 4: 7 districts, population 28;  $\omega(u \leftrightarrow v) = 1$

We can choose any initial partition we like, because we will attempt to improve it in the uncoarsening phase.

In Figure 7, we have randomly chosen the initial partition:

$$D_1 = \{3 + 4, 7\}$$

$$D_2 = \{1 + 2, 5 + 6\}.$$



Table 1: Coarsening coefficients;  
computed from  $\frac{\omega(u)\omega(v)}{\omega(u \leftrightarrow v)}$

Edge	$C$
$1 \leftrightarrow 2$	2
$1 \leftrightarrow 5$	5
$2 \leftrightarrow 3$	6
$1 \leftrightarrow 7$	7
$2 \leftrightarrow 5$	10
$3 \leftrightarrow 4$	12
$3 \leftrightarrow 5$	15
$4 \leftrightarrow 5$	20
$4 \leftrightarrow 6$	24
$5 \leftrightarrow 6$	30
$5 \leftrightarrow 7$	35
$6 \leftrightarrow 7$	42

Table 2: **Maximal matching;**

Vertices 1 and 2, 3 and 4, and 5 and 6 are contracted

Edge	$C$
<b><math>1 \leftrightarrow 2</math></b>	2
$1 \leftrightarrow 5$	5
$2 \leftrightarrow 3$	6
$1 \leftrightarrow 7$	7
$2 \leftrightarrow 5$	10
<b><math>3 \leftrightarrow 4</math></b>	12
$3 \leftrightarrow 5$	15
$4 \leftrightarrow 5$	20
$4 \leftrightarrow 6$	24
<b><math>5 \leftrightarrow 6</math></b>	30
$5 \leftrightarrow 7$	35
$6 \leftrightarrow 7$	42

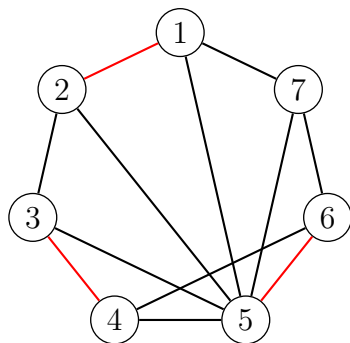


Figure 5: **Maximal matching**

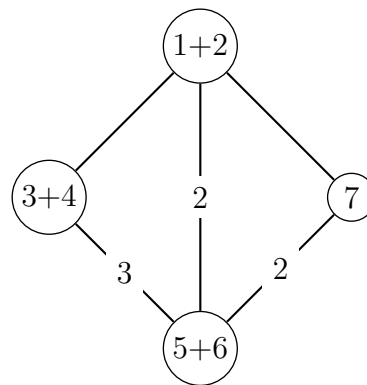


Figure 6: Coarsened graph

Now that we have a partitioned graph, we must project the partition onto the original graph, that is, we must uncoarsen the graph while keeping track of partition membership. Note that we have perfect population balance and the edge cut size is 7. Curious mathematicians would ask, can we do better? When we uncoarsen the graph, we will swap vertices among districts to (hopefully) find a better edge cut.

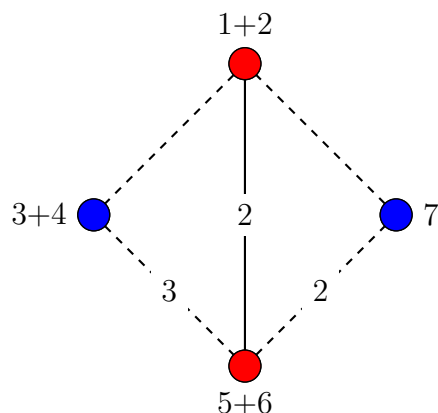


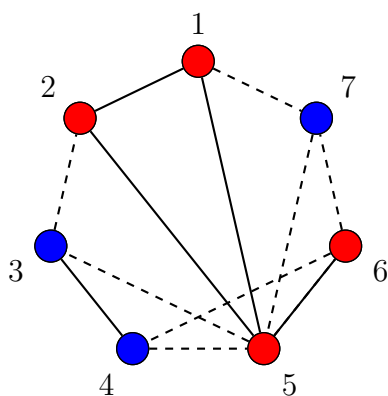
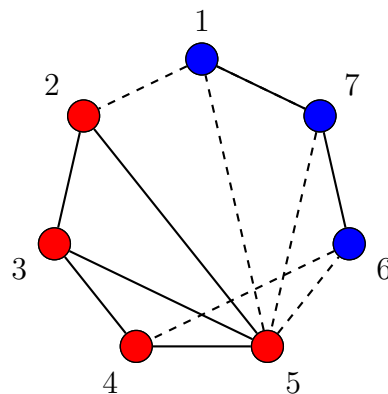
Figure 7: Arbitrary initial partition of the coarsened graph

## 4.2 Uncoarsening

Next, we project the partition onto the original graph, resulting in two districts (shown in Figure 8):

$$D_1 = \{3, 4, 7\}$$

$$D_2 = \{1, 2, 5, 6\}.$$

Figure 8: Partitioned,  
uncoarsened graphFigure 9: Example of local refinement:  
swap (3,4) with (1,6)

In a process called *local refinement*, we swap census units in different districts to find a smaller edge cut. There are  $|D_1| \cdot |D_2| = 3 \cdot 4 = 12$  possible improvements to the edge cut. Sanders and Schulz [12] limit the possibilities by finding negative cycles in the graph, although in our example, we are content with trying every possibility.

If we swap vertices 3 and 4 in  $D_1$  with vertices 1 and 6 in  $D_2$ , then the edge cut size becomes 5, and we maintain perfect population balance. Thus we have improved the edge

cut size from 7 to 5, and our optimal solution is given by the following districts (shown in Figure 9):

$$D_1 = \{1, 6, 7\}$$

$$D_2 = \{2, 3, 4, 5\}.$$

In summary, we have calculated coarsening coefficients of the edges in our graph to greedily build a maximal matching. After choosing an initial bipartition of the graph, we considered swapping vertices in the two districts to find a smaller edge cut while also maintaining population balance. The result is a bipartition with an edge cut size of 5 and equally populated districts.

## 5 Application of Graph Partitioning Algorithm

In terms of redistricting, minimizing the edge cut appears to be correlated with our intuitive idea of compactness. The Census Bureau breaks dense populations into several small census block groups, whereas sparse populations are divided among larger census block groups. Since the edge cut is minimized, we are discouraged from dividing dense populations — they contain many census block groups, and thus many connecting edges. Hence it is common to see partitions where districts cover a single city surrounded by rural area, allowing compactness to be preserved.

Consider the state of Ohio as an example. Ohio contains 9,238 census block groups, making it infeasible to partition by hand; thus we appeal to open source C++ software called KaHIP [12] to partition the state. KaHIP implements the multilevel algorithm with incredible speed. It takes approximately one minute to partition Ohio into 16 districts, where district populations vary at most 1% from the mean. This parameter can be changed, and a longer time may also be specified. Each redistricting map is the product of an independent, pseudo-randomized process.

Figures 10 and 13, identified by their respective edge cut sizes, show the state after 10 minutes of recursive partitioning (the partition with the best edge cut is chosen at the end of the time limit). Figures 11 and 14 show Ohio after 20 minutes of partitioning. Note that the maps with the best and the worst edge cut sizes are not contiguous. The two maps shown in Figures 10 and 11 are contiguous and relatively equal in population; see Table 3 in the appendix for the population counts of the 16 districts. The map in Figure 11 is preferable to the one shown in Figure 10 in the sense that its districts are more equally populated. These maps were chosen to illustrate the possibility of having non-contiguous districts, and how results may vary. Compare these with the 2010 redistricting plan for Ohio in Figure 12.

Although we may specify a smaller population imbalance, the resulting districts may not be as compact, and it is possible to encounter non-contiguous districts such as those in Figures 13 and 14. This provides motivation for modifying the algorithm to consider only contiguous results, and perhaps provide a proof that such a result is always possible to obtain.

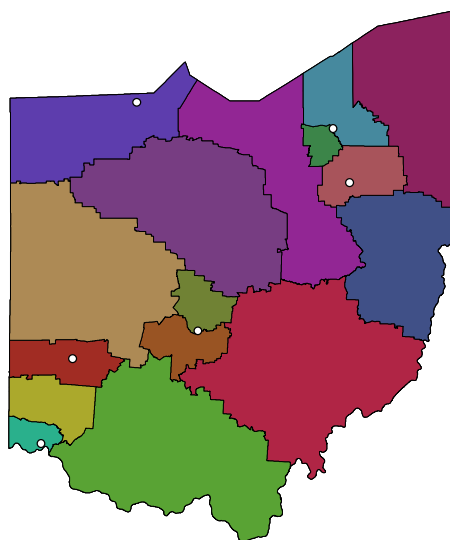


Figure 10: KaHIP partition of Ohio: edge cut size 763

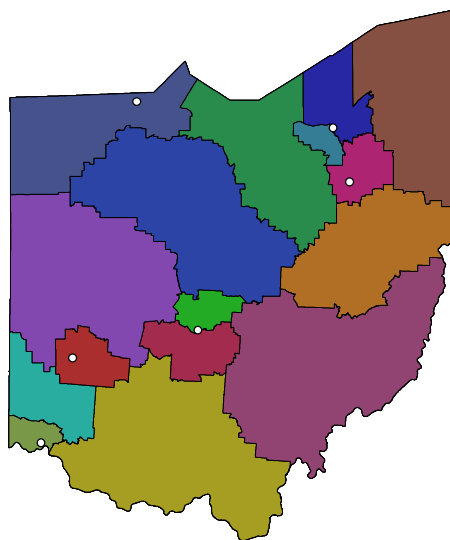


Figure 11: KaHIP partition of Ohio: edge cut size 786

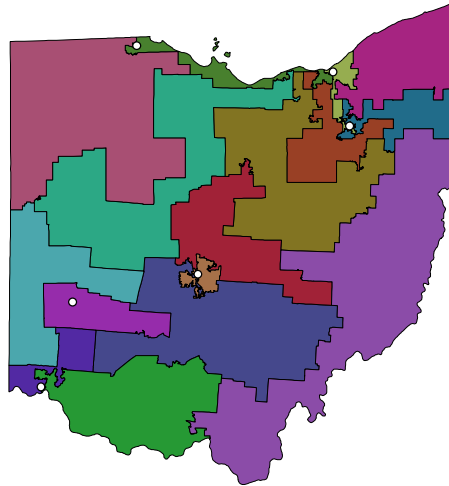


Figure 12: The 16 Ohio Congressional Districts adopted after the 2010 census

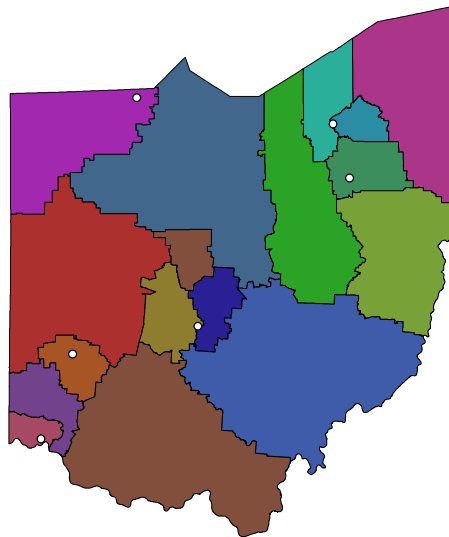


Figure 13: KaHIP partition of Ohio: edge cut size 832; non-contiguous district in brown color

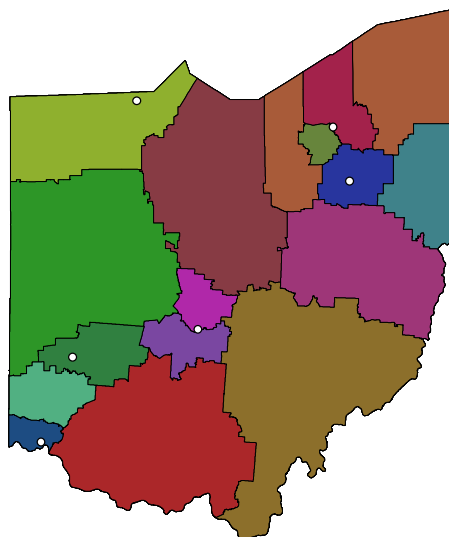


Figure 14: KaHIP partition of Ohio: edge cut size 756; non-contiguous district in rust color

## 5.1 Future Work

Our implementation of the multilevel algorithm uses unit edge weights. If the edges were weighted by distance between census block group centroids, then we would expect more compact districts since minimizing distance would be made the goal of the algorithm. By making more cuts in rural areas, we ensure that dense city populations are not broken up (unless the city is especially populous), a common tactic in gerrymandering. Instead, we minimize the edge cut size to indirectly influence district compactness. In future implementations, we may consider adjusting the edge weights to admit a more realistic approach.

Additionally, census block groups are not the smallest units of population data that the U.S. Census Bureau provides. If we had used Ohio's 365,344 census blocks as opposed to its 9,238 census block groups, then there may have been a smaller population imbalance due to the increased number of possible outcomes. Using census block groups, we have  $16^{9238}$  possible redistricting plans; with census blocks, the number is an even more staggering  $16^{365344}$  (these include non-contiguous redistricting plans). The same population constraints would apply, so intuitively, we would expect a greater number of contiguous redistricting plans.

An undesirable trait of the multilevel algorithm is that it does not guarantee contiguous districts. The redistricting plans are not unique, so it is unclear how a redistricting committee would use the algorithm to choose 'politically fair' districts. The redistricting plans are not unique, as they are a product of pseudo-randomized processes, especially local refinement. Thus it is possible for lawmakers to apply the algorithm to a given state multiple times

and choose whichever plan they prefer. One way to control for this would be to generate a pre-specified number of redistricting plans, and take the district with the best combined population equality and compactness.

We have neglected to consider the ill-defined ‘communities of interest’ that the redistricting process must not break apart. It is possible to modify the algorithm so that these regions are placed in the same district, however federal law does not identify what they are. The addition of other parameters like party affiliation and past election results do not constitute a constitutionally defensible model since the Constitution neither recognizes political parties nor election results. We have erred on the side of simplicity to avoid political complications in the model.

The real strength of using the graph partitioning method in redistricting is that it allows committees to use census population data as the sole parameter in redistricting. This is not only attractive, but constitutionally sound — population change is the purpose of redistricting. In an attempt to create ‘politically competitive’ districts, other models have considered diverse population characteristics, whereas this model treats individuals with the equality that our federal republic requires.

## References

- [1] Steffen Borgwardt, Andreas Brieden, and Peter Gritzmann, *Geometric clustering for the consolidation of farmland and woodland*, The Mathematical Intelligencer **36** (2014), no. 2, 37–44.
- [2] Christopher P. Chambers and Alan D. Miller, *A measure of bizarreness*, 2007.
- [3] Chung-I Chou and Sai-Ping Li, *Taming the gerrymander-statistical physics approach to political districting problem*, Physica A: Statistical Mechanics and its Applications **369** (2006), no. 2, 799–808.
- [4] Hongjiang Chu, Yue Wu, Qiang Zhang, and Yuehua Wan, *Colonial algorithm: A quick, controllable and visible one for gerrymandering*, Information and Automation, Springer, 2011, pp. 424–430.
- [5] Qiang Du, Maria Emelianenko, and Lili Ju, *Convergence of the lloyd algorithm for computing centroidal voronoi tessellations*, SIAM journal on numerical analysis **44** (2006), no. 1, 102–119.
- [6] Jonathan K Hodge, Emily Marshall, and Geoff Patterson, *Gerrymandering and convexity*, The College Mathematics Journal **41** (2010), no. 4, 312–324.
- [7] Zeph Landau and Francis Edward Su, *Fair division and redistricting*, CoRR **abs/1402.0862** (2014).

- [8] Stuart Lloyd, *Least squares quantization in pcm*, Information Theory, IEEE Transactions on **28** (1982), no. 2, 129–137.
- [9] Todd Makse, *Defining communities of interest in redistricting through initiative voting*, Election Law Journal **11** (2012), no. 4, 503–517.
- [10] MATLAB, *version 8.3.0.532 (r2014a)*, The MathWorks Inc., Natick, Massachusetts, 2014.
- [11] Peter Sanders and Christian Schulz, *Engineering multilevel graph partitioning algorithms*, Algorithms–ESA 2011, Springer, 2011, pp. 469–480.
- [12] ———, *Think Locally, Act Globally: Highly Balanced Graph Partitioning*, Proceedings of the 12th International Symposium on Experimental Algorithms (SEA’13), LNCS, vol. 7933, Springer, 2013, pp. 164–175.
- [13] Luke Svec, Sam Burden, and Aaron Dilley, *Applying voronoi diagrams to the redistricting problem*, 2007.
- [14] Attila Tasnádi, *The political districting problem: A survey*, Society and Economy **33** (2011), no. 3, 543–554.
- [15] H Peyton Young, *Measuring the compactness of legislative districts*, Legislative Studies Quarterly (1988), 105–115.



## Appendix

Districts	Fig. 1 (Current)	Fig. 14, NC	Fig. 10, C	Fig. 11, C	Fig. 13, NC
1	721032	740330	700844	726339	723906
2	721031	731213	726937	727006	723555
3	721031	739381	688310	727516	723447
4	721032	739011	732883	720079	713459
5	721031	598895	732987	719522	723358
6	721032	741227	738646	715931	723453
7	721031	741007	724381	717936	710104
8	721032	684027	739826	721050	719008
9	721032	740840	719862	712113	720408
10	721032	714762	737363	724309	722471
11	721032	721594	698530	719456	723351
12	721031	740079	718072	703621	719663
13	721031	740581	739944	720968	713624
14	721032	720698	729112	724235	723349
15	721031	720408	708410	726724	719097
16	721031	711483	689429	718731	723283
Standard Deviation	0.5164	36046.9539	17940.3870	6223.7741	4350.0031
Edge Cut Size	—	756	763	786	832

Table 3: Population counts in proposed redistricting plans for the state of Ohio (C denotes contiguous plans and NC denotes non-contiguous plans)

Note: 2010 census requires approximately 721032 people per district